Online Equivalence Learning Through A Quasi-Newton Method

Hoel Le Capitaine

Online Equivalence Learning Through A Quasi-Newton Method

Hoel Le Capitaine
LINA (UMR CNRS 6241), École Polytechnique de Nantes,
Rue C. Pauc, 44300 Nantes, France.
Email: hoel.lecapitaine@univ-nantes.fr

***Abstract*—Recently, the community has shown a growing interest in building online learning models. In this paper, we are interested in the framework of fuzzy equivalences obtained by residual implications. Models are generally based on the relevance degree between pairs of objects of the learning set, and the update is obtained by using a standard stochastic (online) gradient descent. This paper proposes another method for learning fuzzy equivalences using a Quasi-Newton optimization. The two methods are extensively compared on real data sets for the task of nearest sample(s) classification.**

***Index Terms*—Fuzzy similarity, nearest-neighbor classification, online learning.**

I. Introduction

A. Preliminaries

The concept of metric or similarity measures is fundamental in many pattern recognition problems such as classification and clustering (see e.g. [1], [2]). Depending on the model, the prediction to be made on observations often relies on the evaluation of the comparison between two observations, or one observation and a prototype. More formally, the prediction is a mapping $\mathcal{X} \to \mathcal{Y}$ from the feature space $\mathcal{X}$ to a set of labels, or classes, $\mathcal{Y}$. For instance, instance-based classifiers [3] such as *nearest prototype*, *nearest neighbor*, heavily rely on the metric used. When dealing with unsupervised classification, the comparison between two objects is also very important, e.g. in the k-means and the fuzzy c-means algorithms.

Among the usual metrics that are used, the Euclidean distance is without a doubt the most prevalent measure. However, it is not always adapted to the observations to be compared. Consequently there are a substantial number of distance measures, and finding an appropriate one is a difficult task.

More recently, instead of selecting a metric, people began to learn a parametric metric. A popular choice is the Mahalanobis distance defined by

$$d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^t M (\mathbf{x} - \mathbf{y})$$

where $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ and M is a $p \times p$ matrix, where p is the dimension of $\mathcal{X}$. The goal is to learn M such that the distance between $\mathbf{x}$ and $\mathbf{y}$ is small if they are in the same class, and large if they do not share the same class. There exists many methods to obtain the matrix M, such as convex optimization methods (see e.g. [4], [5]), or simply by estimating the inverse of the covariance matrix of each class.

Another problem comes from the imprecision or the uncertainty of the data. Fuzzy logic theory is now a well founded framework to deal with such imprecisions. Operations of intersection and union can be modeled by triangular norms (t-norm) and triangular co-norms (t-conorms), respectively. Formally, a t-norm is an increasing, associative and commutative mapping $\top : [0,1]^2 \to [0,1]$ satisfying the boundary condition $\top(x,1) = x$ for all $x \in [0,1]$. Alternatively, a t-conorm $\perp$ shares the same properties, except that $\perp(x,0) = x$ for all $x \in [0,1]$, see [6] for additional details.

A key operation in fuzzy logic is the fuzzy implication [7], [8]. Generally speaking, a fuzzy implication function is a mapping $I : [0,1] \times [0,1] \to [0,1]$ holding the following properties

- I is non-increasing in the first variable,
- I is non-decreasing in the second variable,
- $I(0,0) = I(1,1) = 1$, and $I(1,0) = 0$.

Additionally, there are a lot of other properties that can be desired, *e.g.* the confinement principle, see [7], [8] for details. The two classes of fuzzy implication that are generally used are the S-implications, based on a triangular co-norm, and residual implications (R-implications for short) based on triangular norms.

In this paper, R-implication functions are considered, mainly because they hold the confinement principle: $x \leq y$ iff $I(x,y) = 1$, whereas S-implications do not. Given a t-norm $\top$, its corresponding residuum is defined by

$$I_\top(x,y) = \sup_t \{t \in [0,1] | \top(x,t) \leq y\}. \tag{1}$$

In the case of a left-continuous t-norm, the sup can be substituted by the maximum. In the sequel, the following notation is adopted:

- $X = \{x_1, \cdots, x_n\}$ is the (supposed finite) universe of discourse,
- $f_A(x)$, $\forall x \in X$, is the membership function of a fuzzy set A over X.

B. Fuzzy similarity measures

There are several ways to compare fuzzy values or fuzzy quantities. The first one is based on a broad class of measures of equality based on a distance measure which is specified for membership functions of fuzzy sets. The second category

involves set- theoretic operations for fuzzy sets (fuzzy intersection, union, cardinality) [9], [10]. Finally, a third way of defining a similarity measure consists in using logical concepts of fuzzy implication, as first suggested in [11], following the seminal paper of [12].

In this paper, logical-based similarity measures are considered. In particular, one can define a fuzzy similarity measure $\mathcal{S}(A, B)$ between two fuzzy sets A et B by using a residual implication as follows [13]:

$$\mathcal{S}(A, B) = \mathcal{A}_{x \in X} \left\{ \min \Big(I_{\top}(f_A(x), f_B(x)), I_{\top}(f_B(x), f_A(x)) \Big) \right\} \tag{2}$$

An appealing property of this formulation is that one can choose

- any aggregation operator (providing that boundary conditions and monotony are satisfied),
- any (parametric) triangular norm for the residual implication.

In this paper, the aggregation operator that is used is the arithmetic mean. However more sophisticated operators such as the Choquet integral can be used. In the sequel we refer as equivalence learning the operation of learning the parameter, say λ, of a parametric t-norm giving rise to a residual implication. Naturally, learning an equivalence could be much more complex, e.g. if the weights of a weighted arithmetic means are learnt, but this is left for future research.

II. Learning models

Metric or similarity learning methods are now a well studied problem, and give good results in batch settings. However, when the data arrives sequentially, or the number of observations of the data does not allow to load it entirely in memory, it may be interesting to consider online learning methods. Online learning methods, that consider samples one at a time, tend to take a growing importance in the machine learning community [14].

In this paper, an online learning procedure is adopted. In this setting, the algorithm sequentially receives samples, and predicts an output. Once the output is obtained, the algorithm gets a feedback indicating its goodness. Afterwards, parameters can be changed so that the probability of correct output increases in the next step. An appealing property of online algorithms is that they are relatively simple to implement and quickly (*i.e.* with a few number of iterations) provides good performances [15]. Moreover, the learning algorithm does not require to use all the learning set, so that a great improvement in terms of computation is obtained when one considers large datasets. Although convergence is not guaranteed, experimental results show that it is almost always obtained [14].

The main idea of similarity learning based on the relationship and relevance between samples is as follows. Starting from information related to the degree of correspondence that may exist between any two objects, the principle is to consider this relevance to construct a measure of comparison reflecting this correspondence. A correspondence, or relevance, between two objects a and b can be seen as a pairwise function $f(a, b)$ specifying how strongly a and b can be linked. This relationship can be obtained in many ways. For instance, in image annotation, a user may specify that an image representing a beach is not linked to a picture of a plane, whereas there exists a relationship with an image having a sailboard on it. In supervised classification, this link can simply be the representation of classes: two objects belonging to the same class have a strong correspondence. Naturally, obtaining relevance degrees of all possible pairs of observations is not mandatory. This is one of the advantage of this approach: a reduced amount of information is needed for learning.

More formally, one can consider a set $\mathcal{C}$ of relevance degrees between the samples of the dataset. The relevance degrees are specified with *relevance functions*. For simplicity, a relevance function $f : [\mathcal{X} \times \mathcal{X}] \to [0, 1]$ is a function taking two objects as inputs, and output a real value in the unit interval. If $f(a, b) = 1$, then there is a maximum correspondence between a and b, and if $f(a, b) = 0$, it means that the objects are not related with each other, provided the current knowledge of the system.

From this relevance relationships, we seek to build a similarity measure respecting the order induced by the relation: if $f(a, b)$ is greater than $f(a, c)$, then the similarity measure must show a similar behavior:

$$\mathcal{S}(a, b) > \mathcal{S}(a, c) \tag{3}$$

In order to take into account this constraint in our model, we propose to use the following hinge loss function

$$\ell_{\lambda}(a, b, c) = \max \big\{ 0, \mathcal{S}_{\lambda}(a, c) - \mathcal{S}_{\lambda}(a, b) \big\} \tag{4}$$

When the equation (3) holds, the loss function (4) is equal to zero. In contrast, the loss is even more important than the difference between similarities is important. In other words, the more the loss, the greater the correction of the similarity measure.

In batch learning, the standard goal is to minimize the total loss on the entire set of relevance degrees $\mathcal{C}$. This total loss is defined as follows

$$L_{\lambda} = \sum_{(a,b,c) \in \mathcal{C}} \ell_{\lambda}(a, b, c). \tag{5}$$

Instead of a sum over all the elements of the training set, all combinations of $\mathcal{C}$ are considered. If $\mathcal{C}$ is complete, this amounts to considering the sum of $\binom{n}{3}$ loss functions, where n is the size of the training set.

Consequently, even with moderately large datasets, this becomes quickly intractable in practice. In order to tackle this problem, we propose to use an online learning model. In this context, we do not minimize the total loss (5), but use the instantaneous loss, *i.e.* the loss associated to the triplet (a, b, c). In practice, in each iteration of the algorithm, the three objects are selected such that $f(a, b) > f(a, c)$, and the corresponding loss ℓ_{λ} is computed and used for updating.

Now we present two optimization models resulting in different updates of similarity measures: stochastic gradient descent and quasi-newton optimization.

A. Stochastic Gradient Descent

Since the similarity measure is determined by its parameter value λ, its optimal value is searched by using an online learning algorithm based on sequential updates of λ. It leads to find λ such that:

$$\lambda_i = \text{argmin}_\lambda \left(\frac{1}{2}\|\lambda - \lambda_{i-1}\|^2 + \alpha \ell_\lambda(a,b,c) \right), \quad (6)$$

where $\alpha \geq 0$. In other terms, during the optimization process, λ_i is selected in order to obtain a trade-off between minimizing the loss on (a,b,c) and staying quite close to its previous value λ_{i-1}. This trade-off is controlled by the 'aggressiveness' parameter α. Naturally, if α is set to zero, then the optimal λ_i value is equal to λ_{i-1}. In contrast, setting α to a high value imposes an important weight on the loss function. It is clear that when the loss $\ell_\lambda(a,b,c)$ is equal to zero, then the optimal parameter value does not change from the previous iteration, *i.e.* $\lambda_i = \lambda_{i-1}$. Otherwise, the objective function $\mathcal{L}$ is defined by

$$\mathcal{L}(\lambda) = \frac{1}{2}\|\lambda - \lambda_{i-1}\|^2 + \alpha\big(- \mathcal{S}_\lambda(a,b) + \mathcal{S}_\lambda(a,c)\big) \quad (7)$$

The optimal solution with respect to λ is such that the gradient of $\mathcal{L}$, $\partial\mathcal{L}(\lambda)/\partial\lambda$, vanishes:

$$\begin{aligned} \frac{\partial \mathcal{L}(\lambda)}{\partial \lambda} &= \lambda - \lambda_{i-1} - \alpha \frac{\partial\big(\mathcal{S}_\lambda(a,b) - \mathcal{S}_\lambda(a,c)\big)}{\partial \lambda} \\ &= 0 \end{aligned} \quad (8)$$

Let G_i be the gradient value

$$\frac{\partial\big(\mathcal{S}_\lambda(a,b) - \mathcal{S}_\lambda(a,c)\big)}{\partial \lambda}. \quad (9)$$

Therefore, the optimal new value λ is given by

$$\lambda = \lambda_{i-1} + \alpha G_i \quad (10)$$

The corresponding algorithm is described in Algorithm 1 (see [13] for details).

B. Quasi-Newton

An online learning method of parametric fuzzy similarity measures by adopting the Quasi- Newton optimization method [16] is now briefly presented. The advantages of Newton methods are that they allow fast convergence, but they require second derivatives.

Given a starting value λ_0, the sequences of $\lambda(i)$ can be obtained as follows

$$\lambda_{i+1} = \lambda_i - \alpha \frac{G_i}{U_i} \quad (11)$$

where G_i is defined by (9), and U_i is the second derivative. With this formulation, the term U_i is approximated to avoid high computational efforts by the difference equation

$$U_i = \frac{G_i - G_{i-1}}{\lambda_i - \lambda_{i-1}} \quad (12)$$

Algorithm 1 Stochastic Gradient Descent Learning algorithm

function SGD UPDATE
  $i \leftarrow 0$
  $\lambda_i \leftarrow$ initial value
  repeat
    Random selection with replacement of a, b and c in the learning set such that $\mathcal{R}(a,b) > \mathcal{R}(a,c)$.
    if $\ell_\lambda(a,b,c) > 0$ **then**
      $G_i \leftarrow \frac{\partial\left(\mathcal{S}_\lambda(a,b) - \mathcal{S}_\lambda(a,c)\right)}{\partial\lambda}$
      $\lambda_{i+1} \leftarrow \lambda_i + \alpha G_i$
    end if
    $i \leftarrow i + 1$
  until convergence
  return λ
end function

As in the previous method, if the gradient value G_i is equal to zero, then the value of λ remains unchanged. The corresponding algorithm is given in Algorithm 2.

Algorithm 2 Quasi-Newton Learning algorithm

function QN UPDATE
  $i \leftarrow 0$
  $\lambda_{i-1} \leftarrow 0$
  $\lambda_i \leftarrow$ initial value
  repeat
    Random selection with replacement of a, b and c in the learning set such that $\mathcal{R}(a,b) > \mathcal{R}(a,c)$.
    if $\ell_\lambda(a,b,c) > 0$ **then**
      $G_i \leftarrow \frac{\partial\left(\mathcal{S}_\lambda(a,b) - \mathcal{S}_\lambda(a,c)\right)}{\partial\lambda}$
      $U_i \leftarrow \frac{G_i - G_{i-1}}{\lambda_i - \lambda_{i-1}}$
      $\lambda_{i+1} \leftarrow \lambda_i - \alpha G_i / U_i$
    end if
    $i \leftarrow i + 1$
  until convergence
  return λ
end function

C. Implementation details

The update equations (10) and (11) depend on the similarity measure, and therefore of the implication and the aggregation operator. Using a similarity measure therefore requires to be able to derive it with respect to its free parameter λ in order to obtain G_i and U_i.

The performance of the algorithms also depend on the initialization of λ_0. Depending on the measure being used, this initialization can vary. In general, it is recommended to specify a value allowing to obtain a similarity measure approximating the dual similarity measure of the L_1 norm. This leads to use a parameter value of the t-norm such that it is close to the Łukasiewicz t-norm.

It can also happen that the value obtained for λ is not in the domain of definition of the t-norm. In this case, the algorithm

stops and the last correct value of λ is chosen. The value λ can also take particular values of the t-norm. In this case the measure used is the one corresponding to the associated t-norm. For instance, if $\lambda = 0$ for the Schweizer-Sklar t-norm, then the similarity obtained by the product is used, thanks to the continuity around 0 of this t-norm.

Two criteria can be considered for convergence. The most simple is to set a fixed number of iterations. The second one is based on the convergence of the learnt parameter λ. Naturally a criterion such as $\|\lambda_i - \lambda_{i-1}\|^2$ is below a given threshold cannot be used because a zero loss in only iteration implies that $\lambda_i = \lambda_{i-1}$. Therefore, an additional parameter stating the number of no change of λ_i needed for convergence must be specified in this case. Finally, note that both criteria can be used: set a maximum number of iterations, and yes the convergence of λ. This is the setting we used in the experimental part. The number of *no change* of λ for convergence checking is set to 5.

It may happens that data set is unbalanced in the sense that it may exists a lot of samples belonging to a particular concept and a few to others. This case is not handled by our proposition, but a balanced random selection in the algorithms can be used.

III. Numerical experiments

A. Datasets and protocol

In the experiments, four datasets coming from the UCI Machine Learning Repository [17] are used : Iris, Vehicle, Satimage and Segment. First, a brief description of the datasets is given.

The Iris dataset contains 150 observations of 3 kinds of iris flowers with 4 attributes: petal and leaf width and length. Setosa iris flowers are easily discriminated, but Versicolor and Virginica are harder to recognize.

The Vehicle dataset contains 18 features of 846 silhouettes of 4 types of vehicle (Opel, Saab, bus and van). The 18 features are geometrical and order moments.

The third dataset, Satimage, is obtained from a Landsat Multispectral Scanner image. It contains 6435 observations described by 36 attributes. The 36 features are the pixel values of a 3×3 square neighborhood in the four spectral bands. Each observation has a label corresponding to the central pixel. There are 6 different labels, that correspond to a ground category. This dataset is known to present a high generalization error.

The last dataset, Segment, is composed of 2310 observations extracted from 7 outdoor images. Each observation is described by 19 color and geometrical features, and belongs to one of the 7 classes of the problem: brick face, sky, foliage, cement, window, path or grass.

Table I summarizes the characteristics of the four datasets. As can be seen in the table, the considered datasets present a large variety in terms of the number of samples, features and classes. Additionally, a two-dimensional PCA projection (not plotted here for brevity) of the datasets reveals that their distributions also present a large variety of shapes. A method

TABLE I
The four datasets used in the experiments.

Dataset	# Samples	# Features	# Classes
Iris	150	4	3
Vehicle	846	18	4
Satimage	6435	36	6
Segment	2310	19	7

allowing to obtain fuzzy features for each sample from the datasets is now described. Let $X = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ denote a learning set of n observations in $\mathbb{R}^p$ that belong to one of the c discrete labels (classes) $y \in \{1, \cdots, c\}$. In the following, we propose a projection of each sample $\mathbf{x}_i$ into a larger feature space whose dimension is $\mathbb{R}^{c \times p}$. Into this new feature space, the samples are described as fuzzy sets, allowing the use of the aforementioned similarity measures.

Each sample $\mathbf{x}_i$ is projected as follows.

- for each feature $f \in \{1, \cdots, p\}$ and for each label $\ell \in \{1, \cdots, c\}$, compute the corresponding mean μ_j and standard deviation σ_j, where $j = f + (\ell - 1)f$.
- the new (projected) sample $\mathbf{x}'$ of $\mathbf{x}$ is given for each j in $\{1, \cdots, c \times p\}$, by

$$x'_j = f(x_j | \sigma_j, \mu_j)$$

For simplicity, the Gaussian membership function is used. It is defined by

$$f(x|\sigma, \mu) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{13}$$

where μ is the mean and σ is the standard deviation. Naturally, other membership functions (trapezoidal, sigmoidal) can be used as well.

As mentioned earlier, one can choose, or even construct, its own triangular norm, and then write the corresponding implication, that gives a similarity measure. For clarity and brevity, we use in this paper the similarity measure obtained by the Hamacher triangular norm defined by

$$\top_H(x,y) = \begin{cases} \text{Drastic t-norm} & \text{if } \lambda = \infty \\ 0 & \text{if } \lambda = x = y \\ \frac{x\,y}{\lambda + (1-\lambda)\,(x+y-x\,y)} & \text{if } \lambda \in [0, \infty[\\ & \text{and } (\lambda, x, y) \neq (0,0,0) \end{cases}$$

The corresponding residual implication is given by [18]

$$I_H(x,y) = \begin{cases} 1 & \text{if } y \geq x \\ \frac{y(\lambda + x - \lambda x)}{y(\lambda + x - \lambda x) + x - y} & \text{otherwise} \end{cases}$$

The similarity measure is finally obtained by using (2) where $\mathcal{A}$ is the arithmetic mean.

B. Evaluation metric

The quality of a similarity measure is evaluated by the use of a ranking precision measure rather than classification accuracy. Average precision is a commonly used metric in information

retrieval. The mean average precision and the discounted cumulative gain (DCG, not considered here) are frequently used in similarity-based classifiers evaluation [19]. Before defining the mean average precision (mAP), we introduce the precision at k ($prec@k$), which is the proportion of first k relevant samples given the similarity measure, i.e. the top k similar samples. The $prec@k$ for a given sample $\mathbf{x}_i$ is defined by

$$prec@k(\mathbf{x}_i) = \frac{1}{k}\sum_{j=1}^{k} \mathbb{1}\big(y(\mathbf{x}_i) = y(\mathbf{x}_{\sigma(j)})\big)$$

where $\mathbb{1}(A)$ is the indicator function : if proposition A is true then $\mathbb{1}(A) = 1$, 0 otherwise, and $y(\mathbf{x}_i)$ is the true label of $\mathbf{x}_i$. The function $\sigma()$ is a permutation such that $\mathcal{S}(\mathbf{x}_i, \mathbf{x}_{\sigma(1)}) \geq \cdots \geq \mathcal{S}(\mathbf{x}_i, \mathbf{x}_{\sigma(k)})$. The average precision at k is then obtained by averaging the precision at k over all samples

$$AP@k = \frac{1}{n}\sum_{i=1}^{n} prec@k(\mathbf{x}_i).$$

Finally, the level k can vary from 1 to k_{max}, and the mean average precision (mAP) is obtained by averaging $AP@k$ across all k values

$$mAP = \frac{1}{k_{max}}\sum_{k=1}^{k_{max}} AP@k$$

Therefore, the mean average precision measure lies in the unit interval, and the larger the better. Note that setting k_{max} to 1 is equivalent, in terms of classification, to the nearest-neighbor classification rule, but this cannot be generalized to the k_{max} nearest-neighbors classification rule. In the sequel, k_{max} is set to 10.

C. Results

The two algorithms are evaluated on the four presented datasets, and Table II gives the detailed results. There is one main parameter to select in the algorithms, the aggressiveness α. It is chosen in the set {1, 10, 100}, and results for each of these values are reported in the table. For comparison purpose, a baseline performance is also given when using the following fuzzy similarity measure

$$\mathcal{S}(A, B) = 1 - \max_{x \in X} |f_A(x) - f_B(x)|, \qquad (14)$$

which is known to give satisfying results in nearest-neighbor classification [13]. Results obtained with another online similarity learning method where a metric matrix W is learned (OASIS, [15]) are also given. Due to the random part of the algorithms, the learning procedures are executed ten times, and average values are reported in the table. The learning set and testing set are randomly selected such that they are respectively 2/3 and 1/3 of the original set. In this classification application, two objects are considered relevant when they belong to the same class, so that the random selection simply consists in taking two objets from the same class, and another from a different one. A notable consequence in this particular setting of classification is that relevance degrees are binary. Standard deviation, as well as the mean number of iterations needed for convergence are also reported in the table. Note that the maximum number of iterations is set to 2000.

As a general comment, one can see that both algorithms roughly give the same mean average precision. However, one can see differences on each dataset when considering different parameters α, as described as follows.

On the Iris dataset, the SGD method gives slightly better results than the QN method for a given parameter α. When the parameter α is increased, the precision increases whatever the method, and the number of iterations needed for convergence decreases. For a small α value, the number of iterations is approximately equal, whereas it is drastically lower for the QN method with a larger α. Finally, standard deviation, i.e. the stability, of both methods is comparable. An interesting point is that the stability increases when α increases.

On the Vehicle dataset, here again the SGD method gives slightly better results than the QN method for a given parameter α. However, when α increases, the mean average precision decreases for SGD and QN methods. The number of iterations decreases as α increases, and like for the Iris dataset, the number of iterations needed for convergence is roughly equal on both methods for a small value of α, and much lower for the QN method as α increases. According to the table, the stability of the SGD method is better than the stability of the QN method on this dataset.

On the Satimage dataset, the best results are obtained with the SGD method, whatever the aggressiveness parameter α. Like on the Vehicle dataset, decreasing the value of α tends to decrease the mean average precision. Increasing α also reduces the number of iterations needed for convergence for the QN method. However, for the SGD method, this relationship is not clear: increasing α may lead to *fall* into a local optimum. Nevertheless, the number of iterations of the QN method is much lower than the one of the SGD method. Here again, the figures show that the stability of the SGD method is better than that of the QN method.

Finally, on the Segment dataset the SGD method gives the best results for $\alpha = 1, 10$, while the QN method gives the best result for $\alpha = 100$. For this dataset, mean average precision increases as α increases. For the QN method the number of iterations decreases as α increases, whereas for the SGD method, the number of iterations is stable whatever the value of α. More generally, the number of iterations for the QN method is lower than for SGD. Contrary to the other datasets, the stability of the QN method is better than SGD for this dataset.

To conclude on this first experiment, one can say that there is no method undoubtedly better than the other. In terms of mean average precision, the SGD method gives better results. However the QN method allows faster convergence, so that this method could be preferred to handle large datasets, whereas the SGD might be choosen when precision is a key factor. The fact that large values of α leads to a reduced number of iterations is easily understandable because it heavily modifies the value of λ in the update equations (10) and (11). Finally,

TABLE II
RESULTS

Method	α	datasets							
		Iris		Vehicle		Satimage		Segment	
		mAP(%)	iter.	mAP(%)	iter.	mAP(%)	iter.	mAP(%)	iter.
SGD	1	93.93 ± 0.19	1249	67.77 ± 0.01	2000	88.25 ± 0.01	1684	82.66 ± 0.30	1656
	10	94.12 ± 0.05	641	67.64 ± 0.29	1728	88.26 ± 0.01	2000	83.05 ± 0.23	1408
	100	94.29 ± 0.04	778	67.33 ± 0.23	1334	87.88 ± 0.06	1154	83.53 ± 0.34	1978
QN	1	93.47 ± 0.14	1973	67.60 ± 0.05	1792	88.22 ± 0.01	2000	82.46 ± 0.54	1333
	10	94.02 ± 0.13	17	63.04 ± 1.55	13	87.88 ± 0.27	14	83.02 ± 0.08	15
	100	94.25 ± 0.06	6	64.67 ± 0.44	7	87.52 ± 0.28	7	84.14 ± 0.07	6
Baseline (14)		93.05		61.68		81.53		80.31	
OASIS		93.25		64.29		84.89		80.18	

since the value λ changes slowly with the SGD method leads to a good stability of the results, whereas large changes from the update of the QN method decrease the stability.

The second experiment is a detailed study of both algorithms on two datasets: Iris and Vehicle. For both datasets, the following aspects are analyzed:

- the influence of the maximum number of iterations on mean average precision
- the influence of the neighborhood considered on mean average precision
- the influence of the aggressiveness parameter α on the number of iterations needed for convergence or stopping.

The influence of the maximum number of iterations on mean average precision is analyzed as follows. The number of iterations varies from 1 to 8,000, with a step of 1000 iterations. For each different number of iterations, the mAP score is reported using the actual λ value. The results for this experiment are repeated for various values of aggressiveness parameter $\alpha = 1, 10$ and 100.

The influence of the neighborhood considered on mean average precision is studied as follows. The maximum level number k_{max} varies from 1 to 10, and for each value, the mAP score is reported.

Finally, the effect of α over the number of iterations is analyzed as follows. The aggressiveness varies from 1 to 100, and for each value of α the number of iterations needed for convergence is reported. If convergence is not reached within 2000 iterations, the algorithm is stopped. For visualization purpose, α is represented in a logarithmic scale.

1) Iris dataset: The results of the first aspect on the Iris dataset are given in Figure 1, with the SGD method (left) and the QN method (right). As can be seen, the mean average precision globally increases with the number of iterations, whatever the value of α, and whatever the learning algorithm. Comparing the curve —×— ($\alpha = 1$) to the two others —●— and —— (respectively $\alpha = 10$ and $\alpha = 100$) shows that it always provide the worst results. When the number of iterations is low (i.e. the number of updates is low), mAP values are ranked according to the value of α because the larger α, the faster λ is modified. However, setting α to 10 seems to be a better choice when the number of iterations increases. For the QN method, the mAP scores also globally increase when the number of iterations increases. Looking at the curve —×— ($\alpha = 1$) shows that it first gives poorer results than —●— and —— (respectively $\alpha = 10$ and $\alpha = 100$), but greater with a large α ($\alpha = 100$) for a larger number of iterations. The same behavior than the one observed for the SGD method occurs: the mAP score obtained with $\alpha = 100$ grows faster, but at the price of a lower performance than the two others later on.

The results of the two other points that are studied (neighborhood, left and aggressiveness, right) are given in Figure 2. As can be seen on the left figure, the two curves present the same silhouette: the first decrease, and then increase to reach a maximum around 6 neighbors, and finally decrease. For both of them, the value $k_{max} = 6$ seems to be a good choice, whereas choosing $k_{max} = 2$ or 3 gives poor results. On the right, one can see that increasing α leads to a reduced number of iterations. However, the number of iterations for the QN method decreases much faster than the one of the SGD method, confirming the impression of the first experiment.

2) Vehicle dataset: The second dataset is now analyzed. In Figure 3, the mean average precision as function of the number of iterations for the SGD method (left) and the QN method (right) is plotted. As can be seen on the left figure, results with $\alpha = 1$ (—×—) and $\alpha = 10$ (—●—) are comparable. The precision obtained with $\alpha = 100$ (——) is also comparable until the 4000 iterations, where it drastically decrease. On the right, results obtained show that setting α to 1 (—×—) gives the best performance, and $\alpha = 10, 100$ (respectively —●— and ——) give comparable performance.

In the Figure 4-left, the mean average precision as a function of k_{max} is given. For both methods, the lower k_{max}, the better. On the right plot, the same behavior observed on the iris dataset can be seen: increasing α leads to a reduced number of iterations, and the QN method provides faster convergence. Moreover, the number of iterations needed for the SGD method stays quite large.

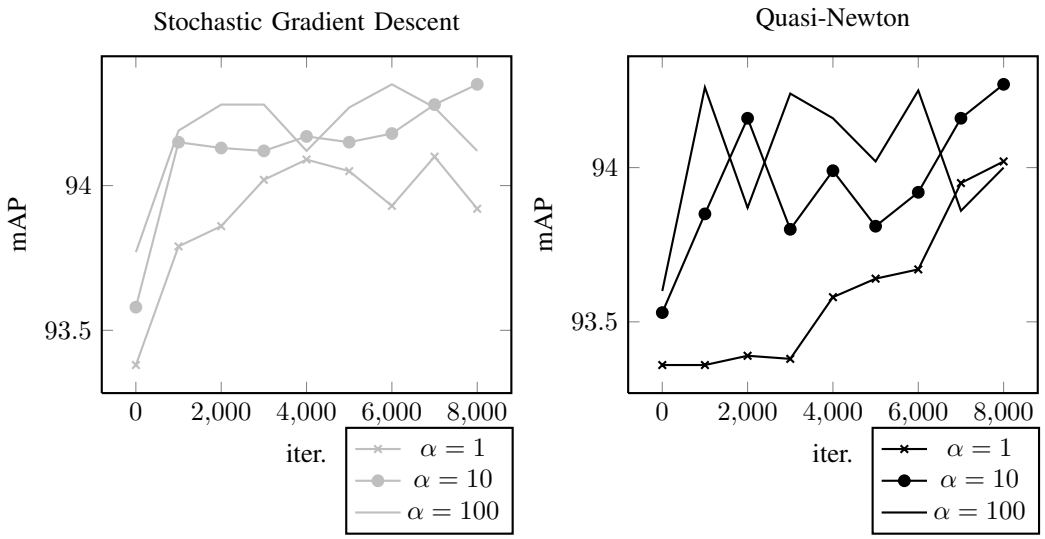

Fig. 1. Mean average precision as a function of the number of iterations for the Iris dataset, with SGD (left) et QN (right) optimization methods.

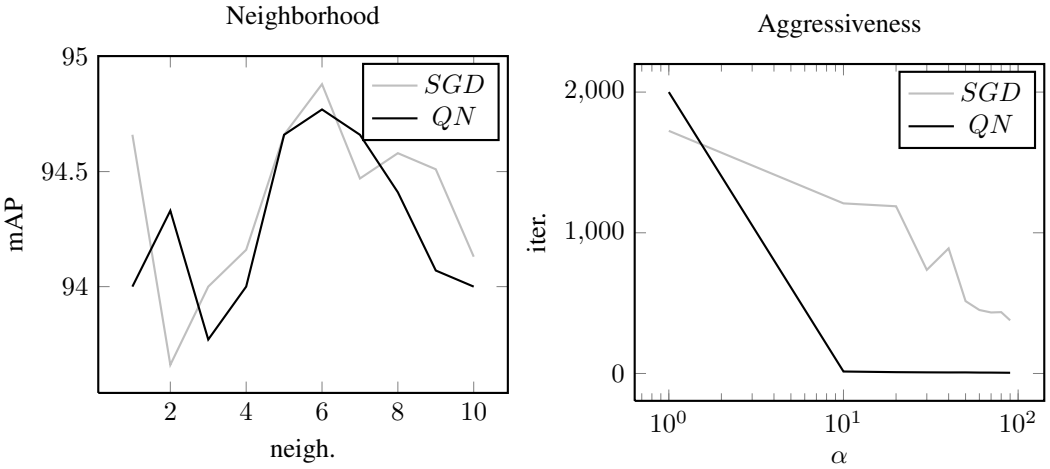

Fig. 2. Evolution of mean average precision as a function of the neighborhood considered (left), and influence of the aggressiveness on the number of needed iterations for convergence on the Iris dataset.

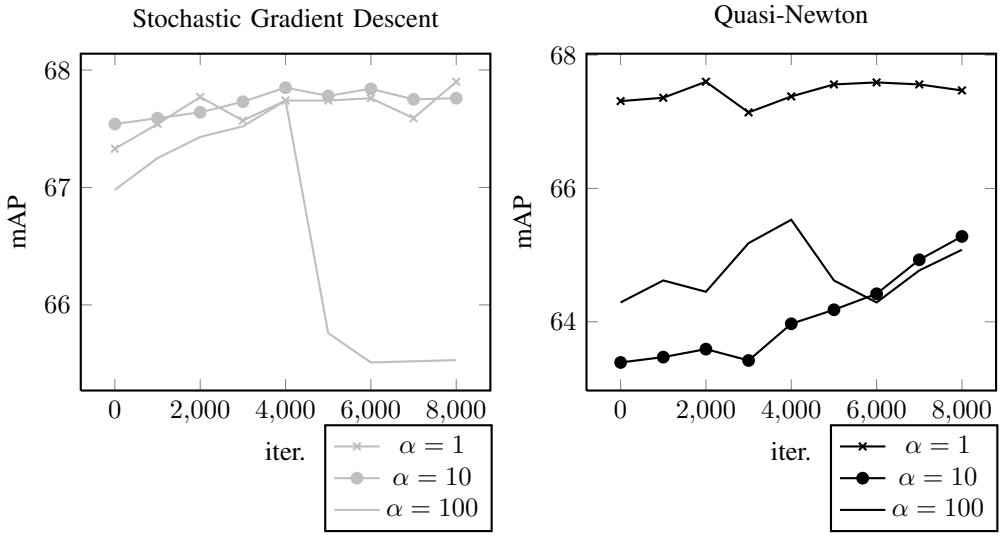

Fig. 3. Mean average precision as a function of the number of iterations for the Vehicle dataset, with SGD (left) et QN (right) optimization methods.

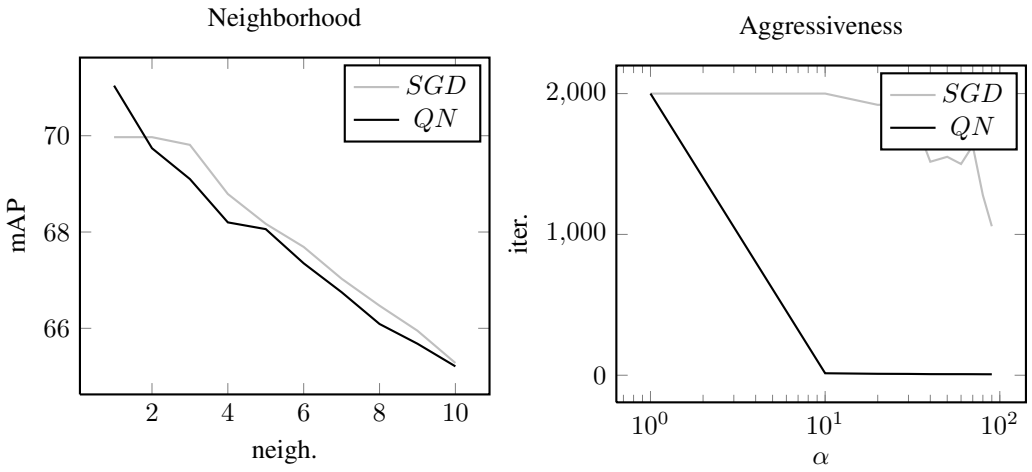

Fig. 4. Evolution of mean average precision as a function of the neighborhood considered (left), and influence of the aggressiveness on the number of needed iterations for convergence on the Vehicle dataset.

IV. Conclusion

In this paper, an optimization method is used for online learning of fuzzy similarity measures defined by fuzzy equivalences. The optimization method is based on Quasi-Newton equation updates, which allows a fast convergence of parameters. Compared to another update algorithm based on stochastic gradient descent, the proposed method exhibits the following advantage. The number of iterations needed is low, so that the method is adapted to the processing of large datasets. However, according to the evaluation metric, the method based on the stochastic gradient descent gives the best results. Eventually, the end-user should use one or another method depending its requirements in terms of precision and the dimension of the dataset. The proposed application is dedicated to classification, but it can also be used for ranking, novelty detection, and so on.

As future line of research, an interesting idea is to use other aggregation function than the arithmetic mean. For instance, weighted means, or Choquet integrals could be used. In this context, individual weights, or coalition weights can be learnt online in the manner of [20], or following the recent work [21]. Another important point is the specification of α, that could change over the iterations, wether by line search [16], a time decay factor or depending on the density in the feature space of the current sample.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[2] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2000.

[3] D. Aha, D. Kibler, and M. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, 1991.

[4] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, June 2009.

[5] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *NIPS*, ser. Advances in neural information processing systems, vol. 15, 2002, pp. 505–512.

[6] M. Grabisch, J. Marichal, R. Mesiar, and E. Pap, *Aggregation Functions*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2009, no. 127.

[7] M. Mas, M. Monserrat, J. Torrens, and E. Trillas, "A survey on fuzzy implication functions," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 6, pp. 1107–1121, 2007.

[8] M. Baczynski and B. Jayaram, *Fuzzy Implications*, ser. Studies in Fuzziness and Soft Computing. Springer, Berlin, 2008, vol. 231.

[9] A. Tversky, "Features of similarity," *Psychological review*, vol. 84, no. 4, pp. 327–352, 1977.

[10] B. Bouchon-Meunier, M. Rifqi, and S. Bothorel, "Towards general measures of comparison of objects," *Fuzzy Sets and Systems*, vol. 84, no. 2, pp. 143–153, 1996.

[11] K. Hirota and W. Pedrycz, "Matching fuzzy quantities," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 6, pp. 1580–1586, 1991.

[12] W. Bandler and L. Kohout, "Fuzzy power sets and fuzzy implication operators," *Fuzzy Sets and Systems*, vol. 4, pp. 13–30, 1980.

[13] H. Le Capitaine, "A relevance-based learning model of fuzzy similarity measures," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, 2012.

[14] S. Shalev-Shwartz, "Online learning: Theory, algorithms, and applications," Ph.D. dissertation, The Hebrew University of Jerusalem, July 2007.

[15] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *Journal of Machine Learning Research*, vol. 11, pp. 1109–1135, 2010.

[16] J.-F. Bonnans, J. C. Gilbert, C. Lemarechal, and C. A. Sagastizabal, *Numerical Optimization: Theoretical and Practical Aspects*. Springer, 2006.

[17] A. Frank and A. Asuncion, "UCI machine learning repository," 2010.

[18] H. Le Capitaine and C. Frélicot, "Towards a unified logical framework of fuzzy implications to compare fuzzy sets," in *13th International Fuzzy Systems Association World Congress, IFSA*, Lisboa, Portugal, 2009.

[19] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti, "Similarity-based classification: Concepts and algorithms," *Journal of Machine Learning Research*, vol. 10, pp. 747–776, 2009.

[20] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2165–2176, 2004.

[21] G. Beliakov, S. S. James, and L. Gang, "Learning Choquet-integral-based metrics for semisupervised clustering," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 3, pp. 562–574, 2011.